Bernstein polynomials in element-free Galerkin method

O F Valencia, F J Gómez-Escalonilla, D Garijo and J L Díez

Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2011 225: 1808 originally published online 16 May 2011
DOI: 10.1177/0954406211401677

Bernstein polynomials in element-free Galerkin method

O F Valencia[1], **F J Gómez-Escalonilla**[2], **D Garijo**[1], and **J L Díez**[3]*
[1]SENER, Ingenieria y Sistemas, Madrid, Spain
[2]Airbus Military, Madrid, Spain
[3]ETSI Aeronáuticos. Universidad Politécnica de Madrid, Madrid, Spain

The manuscript was received on 13 June 2010 and was accepted after revision for publication on 3 February 2011.

DOI: 10.1177/0954406211401677

Abstract: In the recent decades, meshless methods (MMs), like the element-free Galerkin method (EFGM), have been widely studied and interesting results have been reached when solving partial differential equations. However, such solutions show a problem around boundary conditions, where the accuracy is not adequately achieved. This is caused by the use of moving least squares or residual kernel particle method methods to obtain the shape functions needed in MM, since such methods are good enough in the inner of the integration domains, but not so accurate in boundaries. This way, Bernstein curves, which are a partition of unity themselves, can solve this problem with the same accuracy in the inner area of the domain and at their boundaries.

Keywords: meshless, shape functions, Bernstein curves, element-free Galerkin method

1 INTRODUCTION

When numerical methods were needed to solve partial derivative equations, around 1950, several solutions were considered. Three methods were mainly used, based in their simplicity: finite element method [**1**], finite volume method [**2**], and finite difference method [**3**].

A few years later, with the appearance of computers, mainly since 1970s, these principal three numerical methods were improved and programmed to solve all problems found by engineers.

Finally, the complexity of such problems has led to a point in which computers are not enough, even with their increased capacity and improved processors, to solve some problems. In some cases, such problems can be solved, but the time consumed is so long that it is not efficient for common industrial uses. We are talking of problems like crack growth, large deformations, phase transformations, etc. All of them have the same difficulty: finite elements, volumes or differences are rigid enough; so redefinition (remeshing) of the domain is required because if it is not done the solution is not accurate. This means that the base of traditional methods is the limit of the methods.

In the earlier 1990s, some authors [**4**] developed several meshless methods (MMs) or meshfree methods (MF). These are methods in which the approximate solution is constructed entirely in terms of a set of evaluation points, and no elements or characterization of the interrelationship of the evaluation points (meshfree or meshless) are needed to construct the discrete equations. It is then possible to develop discrete equations from a set of evaluation points and a description of the internal and external surfaces of the model. For the latter purpose, a CAD description, such as a model in three dimensions, may be used, so MMs may overcome many of the difficulties associated with meshing for three-dimensional (3D) analyses. Although significant progress in meshing techniques has been achieved, meshing still represents a very daunting task.

*Corresponding author: ETSI Aeronáuticos. Universidad Politécnica de Madrid, Plaza del Cardenal Cisneros 3, 28040 Madrid, Spain.
emails: jesus.lopez.diez@upm.es; jesus.lopez.diez@gmail.com

The earliest of this class of methods was smooth particle hydrodynamics (SPH), developed by Lucy [**5**] and Gingold and Monaghan [**6**]. It was used in the modelization of astrophysical phenomena without boundaries such as exploding stars and dust clouds. Compared to other methods, the rate of publications was very modest for many years, limited mainly to the papers of Monaghan and co-workers [**7**, **8**]. Only recently there have been some contributions to the development of this method.

Based on the use of moving least squares (MLS) approximations, another class of methods have appeared recently. The approximation technique itself was first proposed by Lancaster and Salkauskas [**9**], but it was not until much later that Nayroles *et al.* [**10**] employed the MLS approximants in a Galerkin method. They called their method the diffuse element method (DEM). This method was refined by Belytschko *et al.* [**11**] and the modified method was called the element-free Galerkin method (EFGM). The methods based on MLS approximants are generally more robust and accurate than SPH; however, they are more computationally demanding.

Further information about all MMs can be found in reference [**4**], where other common methods such as meshless local Petrov–Galerkin method are explained. Some of these methods are based on an alternative to MLS, named residual kernel particle method, explained in the previously referred reference.

This new family of methods provides powerful tools for dealing with problems not easily solvable by the finite element methods. Particularly, in aeronautical engineering problems, MMs are used, e.g., in solving simple analysis, such as loaded beams [**11**], thin plates [**12**], or thin shells [**13**]. On the other hand, more specific problems are analyzed, problems dealing with the boundaries of finite element method, where difficulties appear: crack growth in panels [**14**], large deformations, composite plates analysis, etc.

In references [**15**] to [**17**], the behaviour of EFGM when varying internal parameters is analysed. However, in all cases, it can be observed that the main contribution to the numerical error of the method is reached in boundaries of the domain. This is because this method is based on the use of shape functions that conform a Partition of Unity (PU) [**4**] by creating them with MLS that does not fit solution to the boundaries.

Bernstein curves, however, obtain an improved accuracy in boundaries without deteriorating the solution in the inner domain, due to the fact that in boundaries, nodal displacements are equal to nodal parameters (all Bernstein polynomials are zero but extreme polynomials). This way, global error is almost neglected and numerical solution is practically exact. This property is not accomplished by MLS-based EFGM because truncation of shape functions is obtained at boundaries. Besides, some parameters could complicate the selection of the best MLS approximation, such as weight functions, support radius, or the disappearance of order of the polynomial base of the analysis. Finally, from the computing point of view, Bernstein curves are polynomials, which are a PU; so computing them gives low-cost shape functions.

In the following paragraphs, the theoretical background of Bernstein polynomials is introduced. Then, such theoretical basis is applied to EFGM. Finally, two examples of Bernstein polynomial-based EFGM are developed; a first problem analyzes a static case, and the second one shows a dynamic application.

2 BERNSTEIN POLYNOMIALS

Bernstein polynomials are a family of polynomials defined in the following way

$$B_n^i(x) = \binom{n}{i} x^i (1-x)^{n-i} \tag{1}$$

where $x \in [0, 1]$; $i = 0,1,\ldots, n$; n is the order of the polynomial family. Graphically, the family of Bernstein curves of the order n is given as follows (Fig. 1).

Bernstein polynomials are a PU of order 1. This means that the following expression is applicable to Bernstein polynomials

$$\sum_{i=0}^{n} B_n^i(x) \cdot x_i^m = x^m \tag{2}$$

Here, $m = 0, 1$; and x_i are the positions of the points in which Bernstein polynomials are applied. Such locations are considered as those in which maxima are reached for Bernstein polynomials

$$\frac{\mathrm{d}B_n^i(x)}{\mathrm{d}x} = \frac{i - nx}{x(1-x)} B_n^i(x) = 0 \tag{3}$$

$$x_i = \frac{i}{n} \tag{4}$$

Solution (4) is only true when $x \neq 0, 1$, where a maximum is reached, but slope is not 0. In Fig. 2, expression (2) is numerically determined for $m = 0$, 1 and $n = 10$.

For higher orders of the PU and different orders of the Bernstein polynomials, it can be easily checked

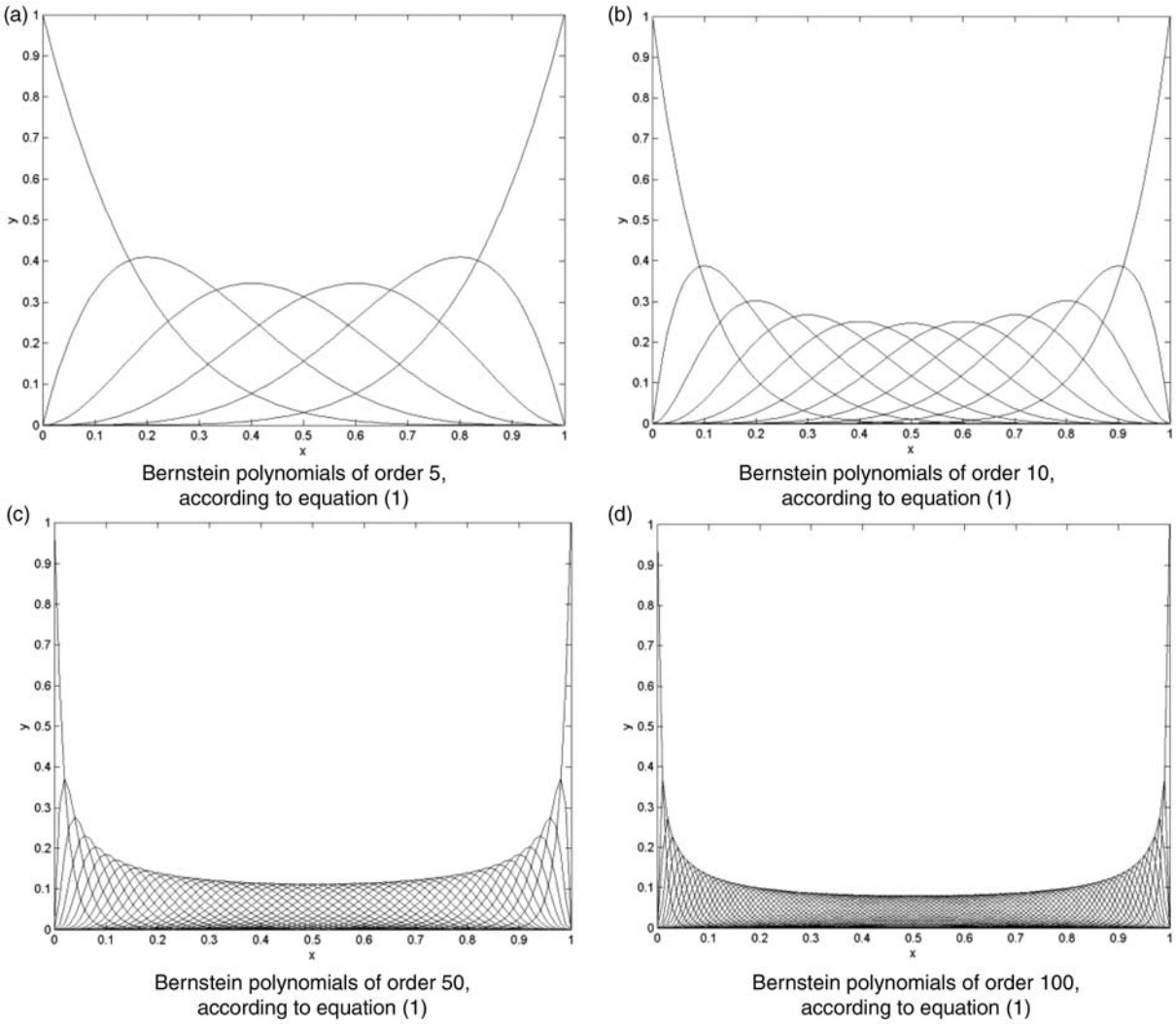

Fig. 1 Bernstein polynomials of (a) order 5; (b) order 10; (c) order 50; and (d) order 100

that only PU condition is reached at extremes of [0,1] interval, or for high values of n.

Another important property of Bernstein polynomials is that their derivatives constitute a partition of nullity (PN); that is

$$\sum_{i=0}^{n} \frac{\mathrm{d}^{\alpha} B_n^i(x)}{\mathrm{d}x^{\alpha}} \cdot x_i^m = 0 \tag{5}$$

As commented in Section 1, Bernstein polynomial-based EFGM allows the solution to reach its exact value at boundaries. This is because at extremes, $B_n^i = 0$, except for $i=0$ and $i=\mathrm{n}$, where $B_n^{0,n} = 1$. Therefore, in enforced boundaries, solution is exact.

3 BERNSTEIN-POLYNOMIAL-BASED EFGM

According to properties shown in Section 2, Bernstein polynomials, therefore, can be used as shape functions for EFGM MM when zeroth or first-order solutions are looked for. Such shape functions are those functions that allow the expression of the solution of the equation in the following way

$$\mathbf{u}^h(\mathbf{x}) = \sum_{k=1}^{N} \phi_k(\mathbf{x}) \cdot \mathbf{u}_k \tag{6}$$

where $\mathbf{u}^h(\mathbf{x})$ is the numerical approximation to the solution; N the number of evaluation points, $_k(\mathbf{x})$ the shape function associated to evaluation point k; and $\mathbf{u}_k$ the nodal parameter associated to the evaluation point k (note that this value is not the nodal solution of $\mathbf{u}^{\mathrm{h}}$ in $\mathbf{x}_k$, which is the main difference between FEM and EFGM).

In the case of an elastostatic analysis, as found in reference [**11**], the general equation of the problem considering a Galerkin implementation is shown in (7)

$$\int_{\Omega} \delta(\mathbf{Lu})^T \mathbf{C}(\mathbf{Lu}) \mathrm{d}\Omega - \int_{\Omega} \delta\mathbf{u}^T \mathbf{b} \mathrm{d}\Omega - \int_{\Gamma_t} \delta\mathbf{u}^T \mathbf{t_0} \mathrm{d}\Gamma - \int_{\Gamma_u} \delta\boldsymbol{\lambda}^T (\mathbf{u} - \mathbf{u_0}) \mathrm{d}\Gamma - \int_{\Gamma_u} \delta\mathbf{u}^T \boldsymbol{\lambda} d\Gamma = 0 \tag{7}$$

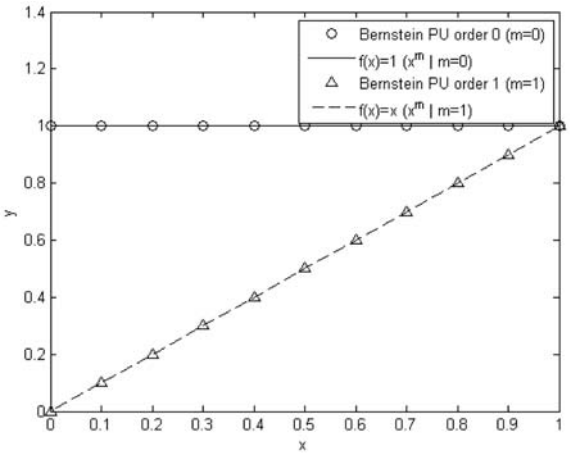

Fig. 2 10th-order Bernstein polynomials, according to equation (1). Straight line for 0th-order PU, according to equation (2), and dashed line for 1st-order PU, according to equation (2)

where Ω is the domain where solution is to be obtained; Γ_t the contour of the domain where loads ($\mathbf{t_0}$) are applied; Γ_u the contour of the domain where solution is imposed to have a predefined value ($\mathbf{u_0}$); $\boldsymbol{\lambda}$ the set of Lagrange multipliers to be considered in order to be able to accomplish with boundary conditions; $\mathbf{C}$ is the structure compatibility matrix [**1**]; and $\mathbf{L}$ the differential operator defined as follows for a 3D analysis

$$\mathbf{L} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ \partial/\partial z & 0 & \partial/\partial x \\ 0 & \partial/\partial z & \partial/\partial y \end{bmatrix} \tag{8}$$

Inserting (6) and (8) in (7), the following equation is reached

$$\begin{aligned} &\sum_{i=1}^{N}\sum_{k=1}^{N} \delta\mathbf{u}_i^T \int_\Omega (\mathbf{L}\phi_i)^T \mathbf{C}(\mathbf{L}\phi_k)\mathrm{d}\Omega \cdot \mathbf{u}_k \\ &\quad - \sum_{i=1}^{N} \delta\mathbf{u}_i^T \int_\Omega \phi_i^T \mathbf{b}\mathrm{d}\Omega - \sum_{i=1}^{N} \delta\mathbf{u}_i^T \int_{\Gamma_t} \phi_i^T \mathbf{t}_0 \mathrm{d}\Gamma - \\ &\sum_{i=1}^{n_\lambda}\sum_{k=1}^{N} \delta\lambda_i^T \int_{\Gamma_u} N_i^T \phi_k \mathrm{d}\Gamma \cdot (\mathbf{u}_k - \mathbf{u}_0) \\ &\quad - \sum_{i=1}^{N}\sum_{k=1}^{n_\lambda} \delta\mathbf{u}_i^T \int_{\Gamma_u} \phi_i^T N_k \mathrm{d}\Gamma \cdot \lambda_k = 0 \end{aligned} \tag{9}$$

Here, $\boldsymbol{\lambda}$ is the column vector of Lagrange multipliers, whose number is n_λ boundary conditions, shown in the following expression

$$\boldsymbol{\lambda} = [N_k \lambda_k] \tag{10}$$

N_k functions of expression (10) are to be considered as those used in reference [**18**]

$$\begin{aligned} N_0(s) &= \frac{s - s_1}{s_0 - s_1} \\ N_1(s) &= \frac{s - s_0}{s_1 - s_0} \end{aligned} \tag{11}$$

Once this point is reached, the way to solve (9) only depends on the shape function. In references [**15**] to [**18**], MLS-based shape functions are obtained; but such functions lead to a solution with a low accuracy in contours due to the lack of PN property at such boundaries.

When Bernstein polynomial-based shape functions are used, PN property of order 1 can be reached at all evaluation points of the domain, including those located in boundaries. Therefore, solution is more accurate than the one obtained by traditional MLS-based EFGM.

For a tridimensional analysis, Bernstein polynomials can be written as follows for a domain in which $x, y, z \in [0, 1]$

$$\begin{aligned} B_{ijk} &= B^{i,j,k}_{n,m,p}(x, y, z) = B^i_n(x) \cdot B^j_m(y) \cdot B^k_p(z) \\ &= \binom{n}{i} x^i (1-x)^{n-i} \cdot \binom{m}{j} y^j (1-y)^{m-j} \cdot \binom{p}{k} \\ &\quad z^k (1-z)^{p-k} \end{aligned} \tag{12}$$

where $n+1$, $m+1$, and $p+1$ are the number of evaluation points in x, y, and z directions.

This way, inserting (12) in (9), Galerkin implementation converts to

$$\begin{aligned} &\sum_{ijk=1}^{N}\sum_{rst=1}^{N} \delta\mathbf{u}_{ijk}^T \int_\Omega (B_{ijk} \cdot B_{rst}) \mathbf{A}_{ijk}^T \mathbf{C} \mathbf{A}_{ijk} \mathrm{d}\Omega \cdot \mathbf{u}_{rst} \\ &\quad - \sum_{ijk=1}^{N} \delta\mathbf{u}_{ijk}^T \int_\Omega B_{ijk}^T \mathbf{b}\mathrm{d}\Omega - \\ &\sum_{ijk=1}^{N} \delta\mathbf{u}_{ijk}^T \int_{\Gamma_t} B_{ijk}^T \mathbf{t}_0 \mathrm{d}\Gamma \\ &\quad - \sum_{i=1}^{n_\lambda}\sum_{rst=1}^{N} \delta\lambda_i^T \int_{\Gamma_u} N_i^T B_{rst} \mathrm{d}\Gamma \cdot (\mathbf{u}_{rst} - \mathbf{u}_0) - \\ &\sum_{jki=1}^{N}\sum_{r=1}^{n_\lambda} \delta\mathbf{u}_{ijk}^T \int_{\Gamma_u} B_{ijk}^T N_r \mathrm{d}\Gamma \cdot \lambda_r = 0 \end{aligned} \tag{13}$$

where

$$\mathbf{A}_{ijk} = \begin{bmatrix} \alpha^i_n(x) & 0 & 0 \\ 0 & \alpha^j_m(y) & 0 \\ 0 & 0 & \alpha^k_p(z) \\ \alpha^j_m(y) & \alpha^i_n(x) & 0 \\ \alpha^k_p(z) & 0 & \alpha^i_n(x) \\ 0 & \alpha^k_p(z) & \alpha^j_m(y) \end{bmatrix} \tag{14}$$

being

$$\alpha_a^b(t) = \frac{b-at}{t(1-t)} \tag{15}$$

So, combining (14) and (15) and inserting in (8)

$$\mathbf{L}\phi_i = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ \partial/\partial z & 0 & \partial/\partial x \\ 0 & \partial/\partial z & \partial/\partial y \end{bmatrix}$$

$$\phi_i = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ \partial/\partial z & 0 & \partial/\partial x \\ 0 & \partial/\partial z & \partial/\partial y \end{bmatrix} B_{ijk} = \mathbf{A}_{ijk} \cdot B_{ijk} \tag{16}$$

(13) can be expressed as a matrix product equation system, according to reference [**18**], in the following way

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} U \\ \lambda \end{Bmatrix} = \begin{Bmatrix} F \\ q \end{Bmatrix} \tag{17}$$

where

$$\begin{aligned} K_{ijk,rst} &= \int_\Omega (B_{ijk} \cdot B_{rst}) \mathrm{A}_{ijk}^T \mathbf{C} \mathrm{A}_{ijk} \mathrm{d}\Omega \\ F_{ijk} &= \int_\Omega B_{ijk}^T \mathbf{b} \mathrm{d}\Omega + \int_{\Gamma_t} B_{ijk}^T \mathbf{t_0} \mathrm{d}\Gamma \\ q_{rst} &= \int_{\Gamma_u} N_i^T B_{rst} \mathrm{d}\Gamma \cdot \mathbf{u_0} \\ G_{ijk} &= -\int_{\Gamma_u} B_{ijk}^T N_r \mathrm{d}\Gamma \end{aligned} \tag{18}$$

In free vibration analysis, (13) is slightly modified to reach the next equation

$$\begin{aligned} &\sum_{ijk=1}^{N} \sum_{rst=1}^{N} \delta \mathbf{u}_{ijk}^T \int_\Omega (B_{ijk} \cdot B_{rst}) \mathbf{A}_{ijk}^T \mathbf{C} \mathbf{A}_{ijk} \mathrm{d}\Omega \cdot \mathbf{u}_{rst} \\ &\quad - \sum_{i=1}^{n_\lambda} \sum_{rst=1}^{N} \delta \lambda_i^T \int_{\Gamma_u} N_i^T B_{rst} \mathrm{d}\Gamma \cdot \mathbf{u}_{rst} - \\ &\sum_{jki=1}^{N} \sum_{r=1}^{n_\lambda} \delta \mathbf{u}_{ijk}^T \int_{\Gamma_u} B_{ijk}^T N_r \mathrm{d}\Gamma \cdot \lambda_r \\ &\quad - \omega^2 \sum_{ijk=1}^{N} \sum_{rst=1}^{N} \delta \mathbf{u}_{ijk}^T \int_\Omega \rho (B_{ijk} \cdot B_{rst}) \mathrm{d}\Omega \cdot \mathbf{u}_{rst} = 0 \end{aligned} \tag{19}$$

whose matritial equation is

$$\begin{bmatrix} K - \omega^2 M & G \\ G^T & 0 \end{bmatrix} \begin{Bmatrix} U \\ \lambda \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{20}$$

where

$$M_{ijk,rst} = \int_\Omega \rho (B_{ijk} \cdot B_{rst}) \mathrm{d}\Omega \tag{21}$$

As can be seen in (13) and in (19), only two parameters are kept from those identified in traditional MLS-based EFGM [**16**]:

1. Number of evaluation points. This is an intrinsic-to-the-MM parameter. When Bernstein polynomials are considered as shape functions, this parameter defines the order of the polynomial. Let N be the number of evaluation points. Bernstein polynomial order is one order lower than this value, that is, $N-1$.
2. Gauss quadrature order. This is an extrinsic-to-the-MM parameter and it is considered only when Gauss quadrature is the scheme used for solving numerical integrals in (13). If another numerical integration method is applied, some other(s) parameter(s) will appear. Hence, better than Gauss quadrature order, it should be considered as a numerical integration scheme with extrinsic parameter.

In the following sections, this theoretical background is developed in order to be applied to some particular cases, and they will be compared to the results obtained with traditional MLS-based EFGM.

4 CASE 1. TWO-DIMENSIONAL ELASTOSTATIC ANALYSIS

The first case to be solved in these pages consists in the Timoshenko's beam, which is the common case that has been solved in literature for MLS-based EFGM [**18**]. In this case, analytical solution is considered for comparing results and to obtain error.

In Fig. 3, a scheme of the domain and boundary conditions is shown.

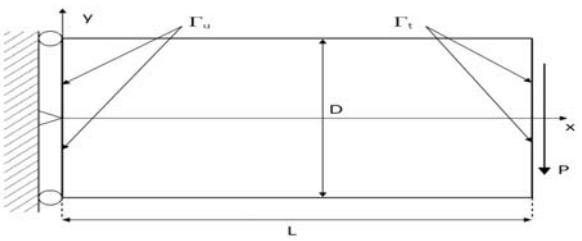

Fig. 3 Numerical case 1. Dimensions, applied loads and boundary conditions for Timoshenko's beam

Governing differential equation

$$\nabla\sigma = \mathbf{0}$$

Boundary conditions are

$$\begin{aligned} &u(x=0,y)=0\\ &v(x=0,y=0)=0\\ &v(x=0,y=\pm D/2)=0\\ &\sigma_{xx}(x=L,y)=0\\ &\sigma_{xy}(x=L,y)=P\left[(D^2/4)-y^2\right]/(2I) \end{aligned} \tag{22}$$

where u is the displacement in x direction; v the displacement component in y direction; σ_{xx} the stress component in x direction; σ_{xy} the shear stress component; P the applied load at boundaries; D the Timoshenko's beam width (in y direction); L the Timoshenko's beam length (in x direction); and t the Timoshenko's beam thickness (in z direction).

Analytical solution for (22) is expressed in (23)

$$\begin{aligned} u(x,y) &= -\frac{Py}{6EI}\left[(6L-3x)x+(2+v)\left(y^2-\frac{D^2}{4}\right)\right]\\ u(x,y) &= -\frac{Py}{6EI}\left[3vy^2(L-x)+(4+5v)\frac{D^2x}{4}+(3L-x)x^2\right] \end{aligned} \tag{23}$$

where E is the Young's modulus of Timoshenko's beam material; v the Poisson's modulus of Timoshenko's beam material; and I the inertia moment of the Timoshenko's beam section.

For MLS-based EFGM solution, Dolbow's results are taken for comparison [**18**]. In such a reference, the error of the numerical solution is taken as the following expression as the rate of convergence in energy

$$\text{energy norm} = \left\{\frac{1}{2}\int_\Omega (\varepsilon_{\text{num}}-\varepsilon_{\text{exact}})^T C(\varepsilon_{\text{num}}-\varepsilon_{\text{exact}})\mathrm{d}\Omega\right\}^{\frac{1}{2}} \tag{24}$$

where C is the characteristic matrix of the material and ε the symmetric gradient of the displacement.

In order to compare the results with reference [**18**], a parameter h is defined as the horizontal distance between the nodes in the model. In reference [**18**], an MLS-based EFGM is used, with a cubic spline shape function, whose $d_{\max}$ value is 2.0 [**16**, **18**]. In reference [**18**], results show a minimum value of energy norm equal to 0.1.

Results obtained for Bernstein polynomial-based EFGM can be seen in Fig. 4.

In Fig. 4, it can be seen that the rate of convergence is lower than that obtained with MLS-based EFGM, as shown in reference [**18**]. Minimum error in displacements at end for MLS-based EFGM, as shown in reference [**18**], is 0.01 per cent when 175 nodes are considered. However, when 10 nodes are used, reference [**18**] shows an error of 8.7 per cent.

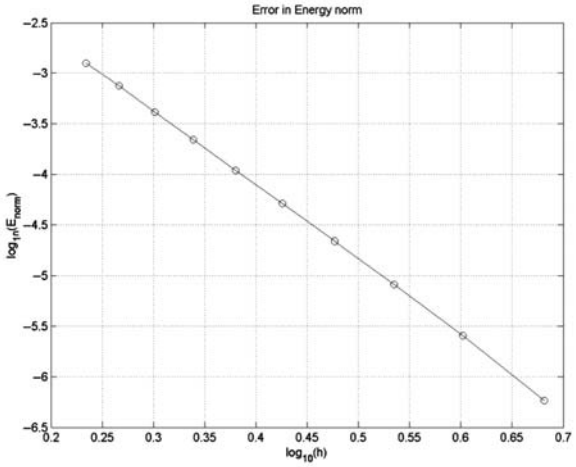

Fig. 4 Rate of convergence in displacements for Bernstein polynomial-based EFGM

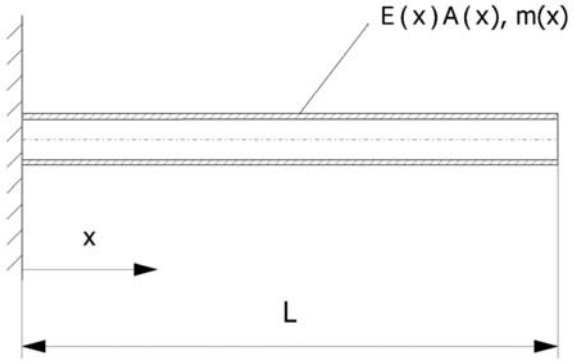

Fig. 5 Numerical case 2. Dimensions and mechanical properties for an axially loaded beam in vibration analysis

When Bernstein polynomial-based EFGM is applied, exact solution is reached with a low number of nodes.

5 CASE 2. ONE-DIMENSIONAL VIBRATION ANALYSIS

A second case is to be solved in these pages. A one-dimensional (1D) vibration problem as shown in reference [**19**], where MLS-based EFGM is considered to obtain the numerical solution, and where analytical expression is evaluated.

In Fig. 5, a scheme of the domain and boundary conditions is shown.

Governing differential equation is

$$m\frac{\partial^2 u}{\partial t^2} - EA\frac{\partial^2 u}{\partial x^2} = 0 \tag{25}$$

The boundary conditions are

$$\begin{aligned} u(0) &= 0\\ u_{,x}(L) &= 0 \end{aligned}$$

Table 1 Maximum error (logarithmic values) in three first natural frequencies

Number of evaluation points	MLS-based ($d_{max} = 2.0$)	MLS-based ($d_{max} = 6.0$)	Bernstein-based
$n = 11$	−7	−11	−17
$n = 21$	−9	−14	Exact
$n = 31$	−10	−16	Exact
$n = 41$	−11	−17	0.5

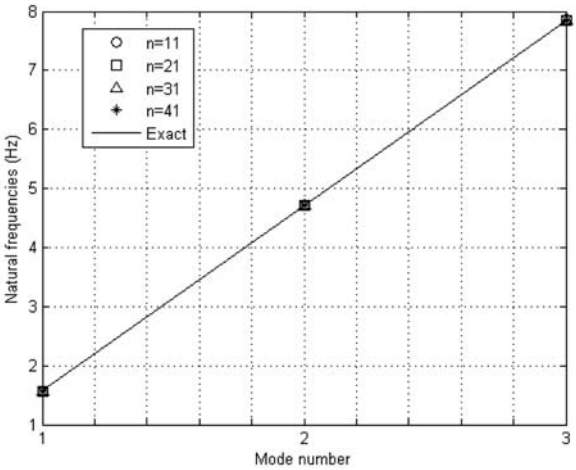

Fig. 6 Natural frequencies for MLS-based EFGM, $d_{max} = 2.0$

The analytical solution for the natural frequencies of (25) is expressed in (26), as obtained from reference [**19**]

$$\omega_n = \frac{\pi}{2}(2n-1)\sqrt{\frac{EA}{m}}\ n = 1,\ 2 \tag{26}$$

The error considered is the rate of natural frequencies, comparing numerical results of natural frequencies, shown as num, and those taken from (26), marked as exact solution

$$E_n = \frac{\left|\omega_n^{\text{num}} - \omega_n^{\text{exact}}\right|}{\omega_n^{\text{exact}}} \tag{27}$$

The error is better shown as the $\log(E_n)$ in Table 1, where solutions for MLS-based (Figs 6 and 7) shape functions and Bernstein (Fig. 8) polynomial-based shape functions are considered. For MLS-based ones, two options have been obtained, one with $d_{max} = 2.0$ (Fig. 6) and another one with $d_{max} = 6.0$ (Fig. 7).

From previous figures, it can be pointed out that, in general, increasing the number of evaluation points improves the accuracy of results in several orders of magnitude. This fact is not effective when Bernstein polynomials are considered to obtain shape functions

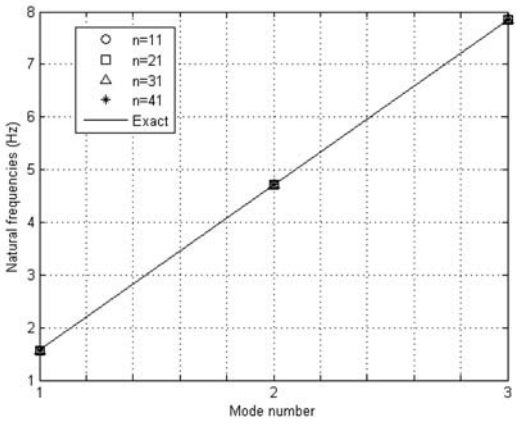

Fig. 7 Natural frequencies for MLS-based EFGM, $d_{max} = 6.0$

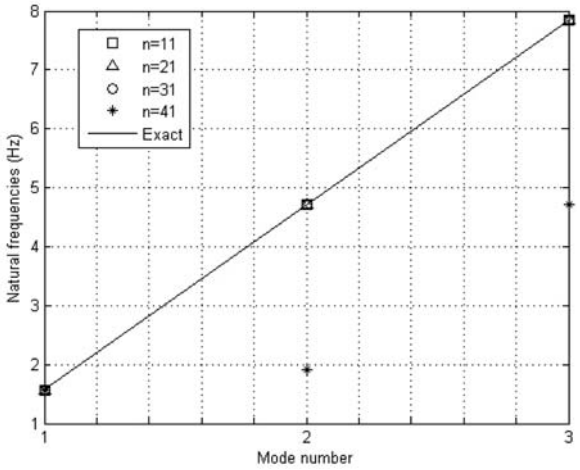

Fig. 8 Natural frequencies for Bernstein polynomial-based EFGM

mainly because a problem of truncation error occurs: when the number of evaluation points is increased, the order of the polynomial is raised too. Therefore, when a certain order is reached, the truncation error appears and the error is increased. It is clear that a number of evaluation points higher that 40 is not adequate because of that for Bernstein polynomial-based EFGM. However, for a high number of evaluation points (higher than 40), if no truncation of decimal positions could be achieved, the solution would be better than those coming from a lower number of evaluation points.

If computer-aided analysis is used, when the number of evaluation points is below 30, Bernstein polynomial-based EFGM is a more accurate solution than MLS-based one, and such a solution can be considered as the exact (the order of the error is below −20 for the three first natural frequencies).

6 CONCLUSIONS

When it is intended that a numerical solution has to be obtained for a structural analysis, elastostatical or dynamical problems, several methods are applicable. If EFGM MM is considered, a shape function is needed. This shape function can be obtained by several ways.

Typically, MLS-based method is used. However, a complicated and time-consuming algorithm is required in MLS-based EFGM. Therefore, Bernstein polynomials, which constitute a PU of order 0 and, in case evaluation points are uniformly distributed along domain of integration, of order 1 too, can be an alternative and easy-computing way of obtaining and evaluating such shape functions.

When analytical solution is a zeroth-order or first-order one, numerical solution is much more accurate than MLS-based EFGM, and analytical solution is exactly obtained with numerical one (except of truncation errors).

ACKNOWLEDGEMENTS

This research has been partially financed by the Spanish Ministry of Science and Technology through the DPI2005-08276 research program. The authors also thank the collaboration of SENER and Airbus Military.

REFERENCES

1 **Zienkiewicz, O. C.** and **Taylor, R. L.** *The finite element method. The basis*, 5th ed, 2000 (Butterworth – Heinemann, Oxford).
2 **LeVeque, R.** *Finite volume methods for hyperbolic problems*, 2002 (Cambridge University Press, Cambridge).
3 **Morton, K. W.** and **Mayers, D. F.** *Numerical solution of partial differential equations: an introduction*, 2005 (Cambridge University Press, Cambridge).
4 **Fries, T. P.** and **Matthies, H. G.** Classification and overview of meshfree methods (Informatikbericht-Nr. 2003-03), 2004 (Technische Universität Braunschweig, Brunswick).
5 **Lucy, L. B.** A numerical approach to the testing of the fission hypothesis. *Astron. J.*, 1977, **82**(12), 1013–1024.
6 **Gingold, R.** and **Monaghan, J.** Smoothed particle hydrodynamics: theory and application to non-spherical stars. *R. Astron. Soc.*, 1977, **181**, 375–389.
7 **Monaghan, J. J.** Why particle methods work. *SIAM J. Sci. Stat. Comput.*, 1982, **3**(4), 422–433.
8 **Monaghan, J. J.** An introduction to SPH. *Comput. Phys. Commun.*, 1988, **48**, 89–96.
9 **Lancaster, P.** and **Salkauskas, K.** Surfaces generated by moving least squares methods. *Math. Comput.*, 1981, **37**, 141–158.
10 **Nayroles, B., Touzot, G.,** and **Villon, P.** Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput. Mech.*, 1992, **10**, 307–318.
11 **Belytschko, T., Lu, Y. Y.,** and **Gu, L.** Element-free Galerkin methods. *Int. J. Numer. Methods Eng.*, 1994, **37**, 229–256.
12 **Krysl, P.** and **Belytschko, T.** Analysis of thin plates by the element free Galerkin method. *Comput. Mech.*, 1995, **17**, 26–35.
13 **Krysl, P.** and **Belytschko, T.** Analysis of thin shells by the element free Galerkin method. *Int. J. Solids Struct.*, 1996, **33**(22), 3057–3080.
14 **Rao, B. N.** and **Rahman, S.** An efficient meshless method for fracture analysis of cracks. *Comput. Mech.*, 2000, **26**, 398–408.
15 **Valencia, O. F., Urbinati, F., Gómez-Escalonilla, F. J.,** and **López, J.** Weight functions analysis in elasto-statics problems for meshless element-free Galerkin method. *Lisbon: III Eur. Comput. Mech.*, 2006, 531–536.
16 **Valencia, O. F., Gómez-Escalonilla, F. J.,** and **López, J.** Influence of selectable parameters in EFGM. 1D bar axially loaded problem. *Proc. IMechE Part C: J. Mechanical Engineering Science*, 2008, **222**(11), 1621–1633.
17 **Valencia, O. F., Gómez-Escalonilla, F. J.,** and **López, J.** Influence of selectable parameters in EFGM. 1D beam in bending problem. *Proc. IMechE Part C: J. Mechanical Engineering Science*, 2009, **223**(7), 1579–1590.
18 **Dolbow, D.** and **Belytschko, T.** An introduction to programming the meshless element free Galerkin method. *Arch. Comput. Methods Eng.*, 1998, **5**(3), 207–241.
19 **Timoshenko, S.** *Vibration problems in engineering*, 2nd ed, 1973 (D. Van Nostrand Company, Inc, New York).